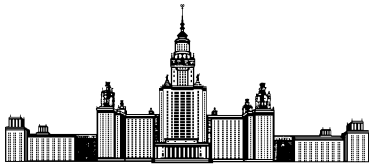
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ОТЧЁТ

«Ансамбли алгоритмов. Веб-сервер. Композиции алгоритмов для решения задачи регрессии.»

Задание №3 по курсу «Практикум на ЭВМ»

Выполнил:

студент 3 курса 317 группы

Зиннуров Артём Ринатович

Научный руководитель:

к.ф-м.н.

Китов Виктор Владимирович

Москва, 2021

Содержание

1 Введение

В данной работе приведены результаты экспериментов на датасете данных о продаже недвижимости **House Sales in King County, USA** - регрессионная задача предсказания стоимости жилья. При решении поставленных задач используются различные ансамбли алгоритмов, реализованные автором отчёта.

2 Предобработка данных

Проведём предобработку имеющихся данных.

1. Посмотрим на типы признаков в представленных данных. Имеем 21 признак, из которых 20 признаков - числовые (`int64`, `float64`), а 1 признак *"date"* типа `object`. Переведём этот признак в тип даты (`datetime64[ns]`), а после из этого признака сделаем 3 целочисленных признака - день, месяц и год (*"day"*, *"month"*, *"year"*) соответствующей даты. Удалим исходный признак *"date"*.

2. В данных нет пропусков, поэтому не приходится думать, чем их заполнять.

3. Удалим из данных признак *"id"*, так как он способствует переобучению.

4. Сформируем из признака *"price"*, представляющего собой предсказываемое значение, целевой вектор. Удалим этот признак из данных.

5. Переведём данные в `numpy.ndarray`.

6. Разделим данные на обучение и контроль с помощью функции *train_ test_ split()* из библиотеки **sklearn**.

3 Исследование поведения алгоритма случайный лес

3.1 Дизайн эксперимента

Исследуем поведение алгоритма **случайный лес**. Изучим зависимость **RMSE** на отложенной выборке и время работы алгоритма в зависимости от следующих факторов:

- количество деревьев в ансамбле;
- размерность подвыборки признаков для одного дерева;
- максимальная глубина дерева (с дополнительным случаем неограниченной глубины).

При реализации алгоритма **случайного леса** использовался метод стохастического ансамблирования **bagging** (подвыборки обучающей выборки "с возвращением") для повышения разнообразия с помощью рандомизации.

Кроме параметров функции *__init__()*, описанных в шаблоне, отдельно обрабатываем параметры `random_state`, `eps_val` и `continuity`, которые вычленяются из аргумента `trees_parameters`. `random_state` (значение по умолчанию равно 42) используется для воспроизводимости результатов экспериментов, `eps_val` (значение по умолчанию равно None) в соответствии с псевдокодом из лекций, используется для отбора алгоритмов с неудовлетворительными значениями функционала ошибки - параметр устанавливает порог на максимально допустимое значение функционала ошибки на валидационных данных и игнорируется при их отсутствии. При этом если задать слишком большое значение этого параметра, все базовые алгоритмы будут забракованы. Поэтому для предотвращения зацикливания `n_estimators` - это именно кол-во итераций цикла, а не кол-во фактически используемых базовых алгоритмов. При предсказании в ситуациях, когда список базовых алгоритмов пуст, будет выброшено исключение, описывающее ошибку. Но в дальнейших экспериментах мы не будем пользоваться этой функциональностью, описанной в лекционном псевдокоде. `continuity` (значение по умолчанию равно False) необходимо для вывода предсказа-

ний для всех значений кол-ва деревьев, меньших и равных заданного. Необходимо для того, чтобы не с нуля переобучать модель для каждого значения кол-ва деревьев.

Для экспериментов выбраны следующие значения гиперпараметров:

- количество деревьев в ансамбле: 1, 5, 10, 20, 50, 100, 200, 300, 500;
- размерность подвыборки признаков для одного дерева: 0.1, 0.3, 0.5, 0.8, 1.0;
- максимальная глубина дерева: 1, 3, 5, 7, 10, 15, None (неограниченная глубина).

3.2 Результаты эксперимента

Общее время обучения на выбранных выше параметрах составило 22 минуты.

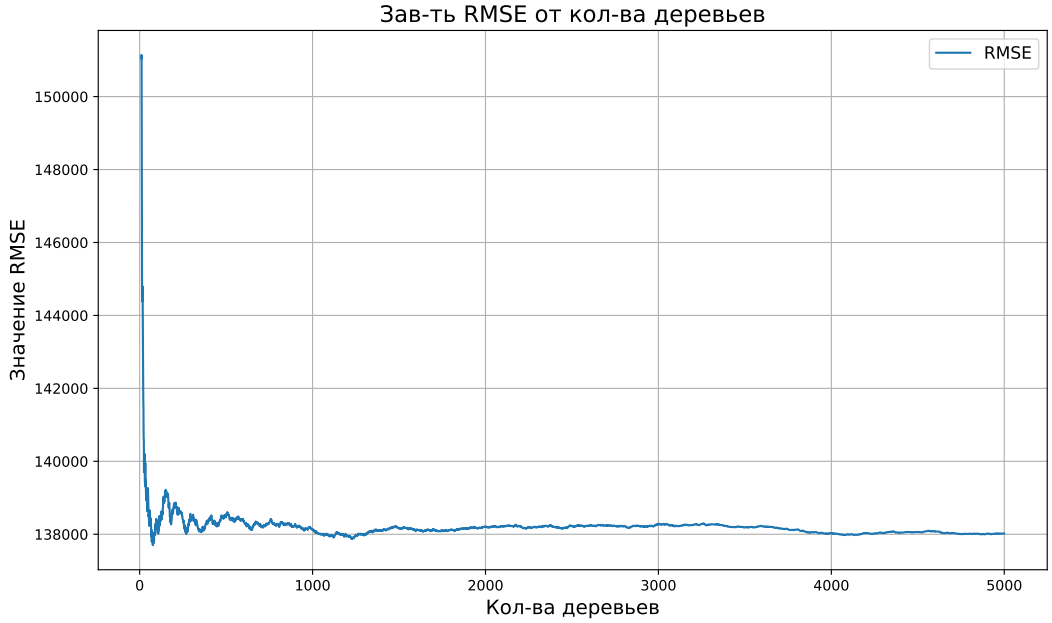


Рис. 1: Зав-ть **RMSE** от кол-ва деревьев. Как видно из графика даже при таком большом кол-ве деревьев в композиции (5000) переобучение всё равно не наблюдается. Объясняется это разложением **BVD** в шум (который зависит только от данных), сдвиг и разброс: при простом голосовании сдвиг не меняется, а разброс уменьшается при условии некоррелированности базовых алгоритмов (об этом подробнее в выводе). В данном случае максимальная глубина дерева не ограничена, а объём используемого признакового пространства - 1/3 (значения по умолчанию). Для лучшего осознания масштабов на интересующих нас значениях значения **RMSE** выводятся для кол-в деревьев, начиная с 10.

Здесь и далее ограничимся 500 деревьями в композиции для ускорения постановки экспериментов.

Также на приводимых ниже рисунках зависимостей будут присутствовать графики для разных значений параметров, зависимость от которых в данном рисунке не исследуется. Это сделано для более детального исследования зависимостей, которые могут наблюдаться только при определённых параметрах, для большей наглядности и информативности графиков.

На всех графиках будет использовано значение доли подвыборки признаков, равное 1.0. То есть при построении очередного дерева будем использовать всю мощь, весь объём имеющихся данных, что на первый взгляд (и только на первый взгляд) кажется довольно логичным.

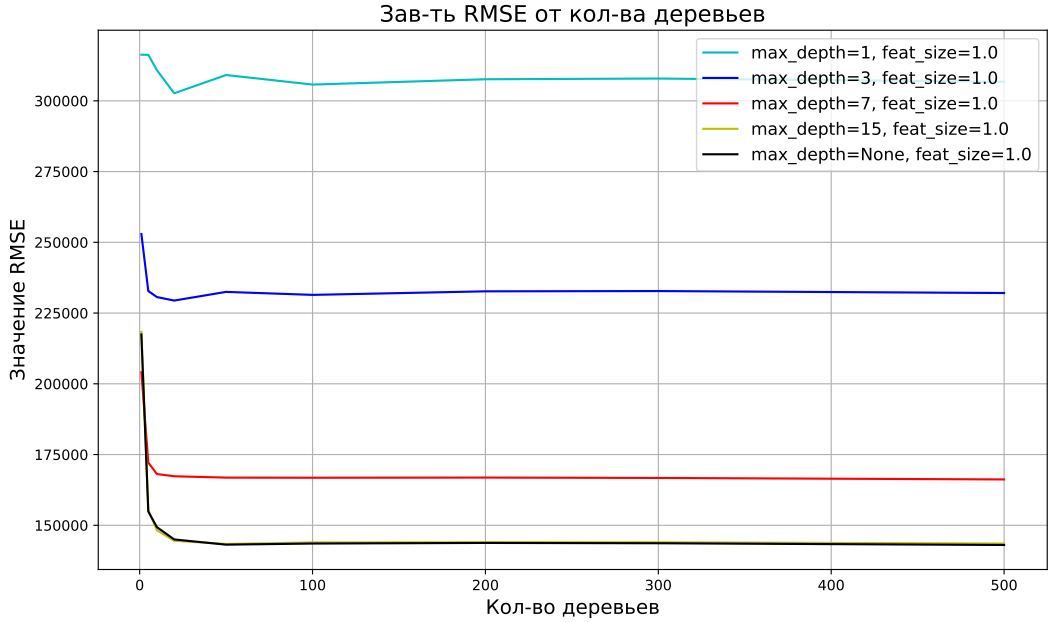


Рис. 2: Зав-ть **RMSE** от кол-ва деревьев. В данном случае приведены графики при доли признаков, равной 1.0, и разных значениях максимальной глубины деревьев. Как видно из рисунка, **RMSE** уменьшается и выходит на плато при всех значениях максимальной глубины. При этом видны некоторые особенности "решающих пней"(`max_depth` $= 1$) и деревьев с `max_depth` $= 3$ - минимум **RMSE** при небольших кол-вах деревьев (деревья быстро начинают коррелировать из-за их архитектурной бедности). На графике также видно, что **RMSE** уменьшается при увеличении максимальной глубины (об этом подробнее на Рис. 4).

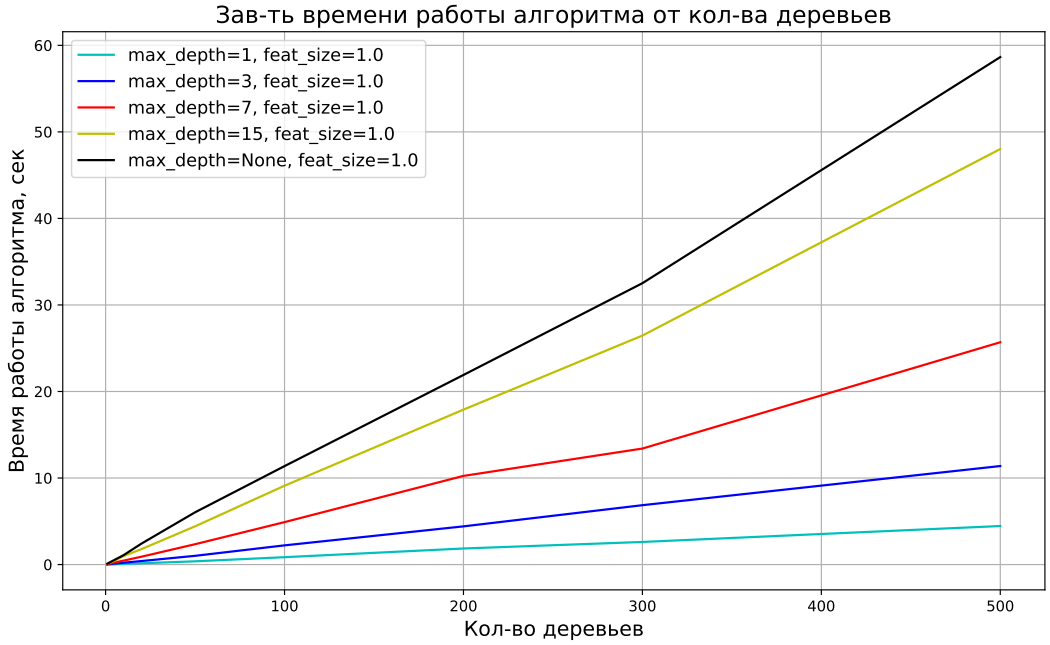


Рис. 3: Зав-ть времени работы алгоритма от кол-ва деревьев. Приведены графики при доле признаков, равной 1.0, и разных значениях максимальной глубины деревьев. Как и предполагалось, время работы увеличивается с ростом кол-ва деревьев (каждое дерево - один базовый алгоритм, который нужно обучить). При этом также видно увеличение времени работы с увеличением максимальной глубины деревьев, что тоже вполне ожидаемо (об этом подробнее на Рис. 5).

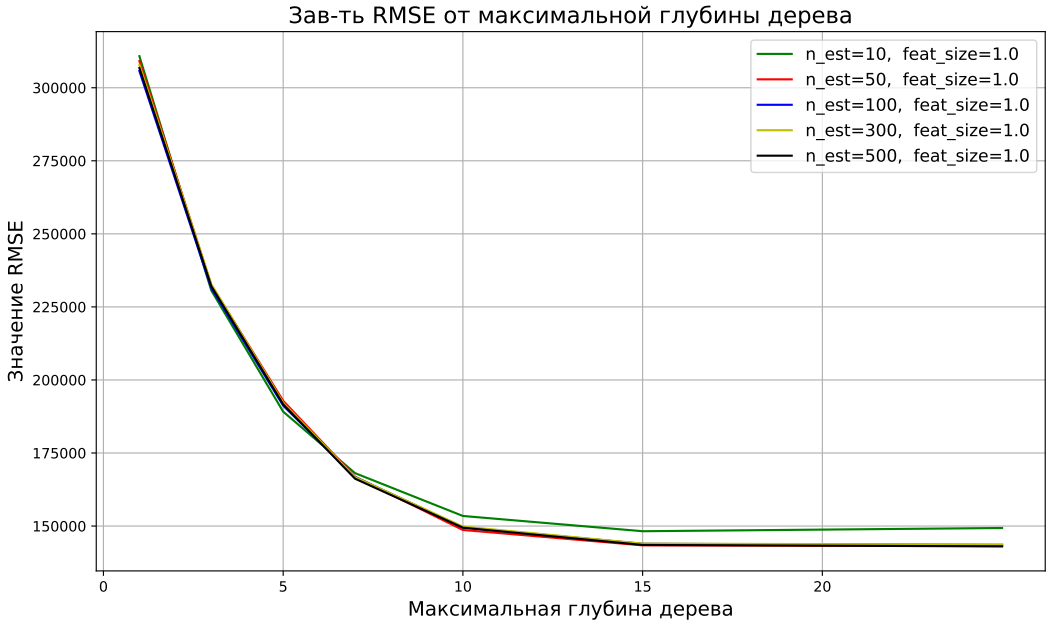


Рис. 4: Зав-ть **RMSE** от максимальной глубины дерева. Приведены графики при доле признаков, равной 1.0, и разных значениях кол-ва деревьев. Видно, что увеличение глубины дерева приводит к уменьшению **RMSE**.

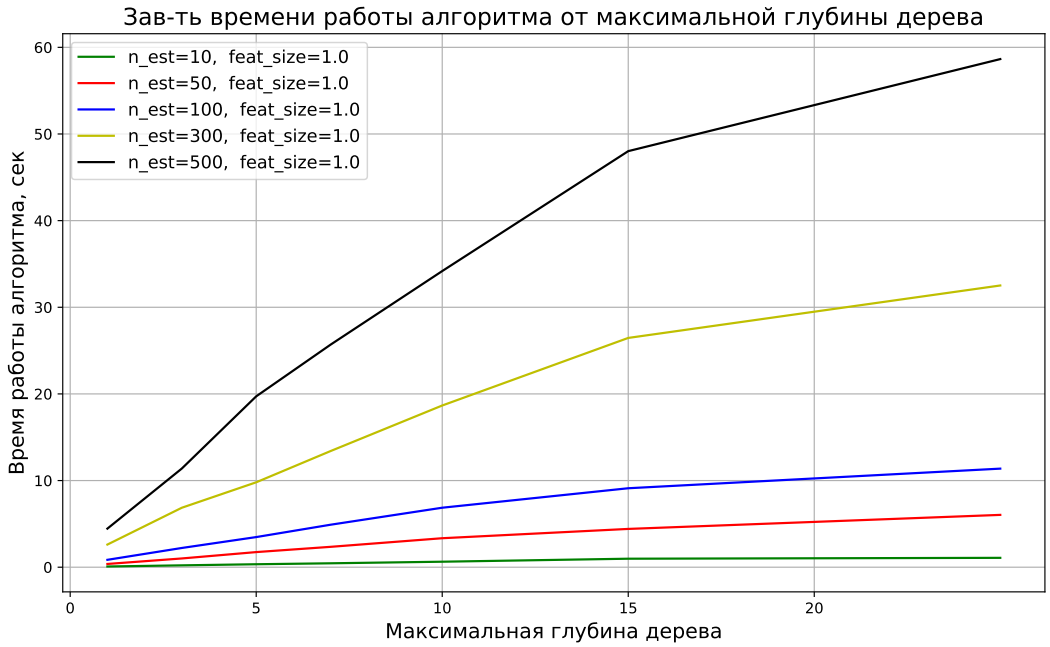


Рис. 5: Зав-ть времени работы алгоритма от максимальной глубины дерева. Приведены графики при доле признаков, равной 1.0, и разных значениях кол-ва деревьев. Время работы увеличивается при увеличении максимальной глубины дерева. Объяснятся это увеличением объёмов вычислений при построении более глубокого дерева. Также, как уже было отмечено выше, увеличение кол-ва деревьев приводит к увеличению времени работы алгоритма.

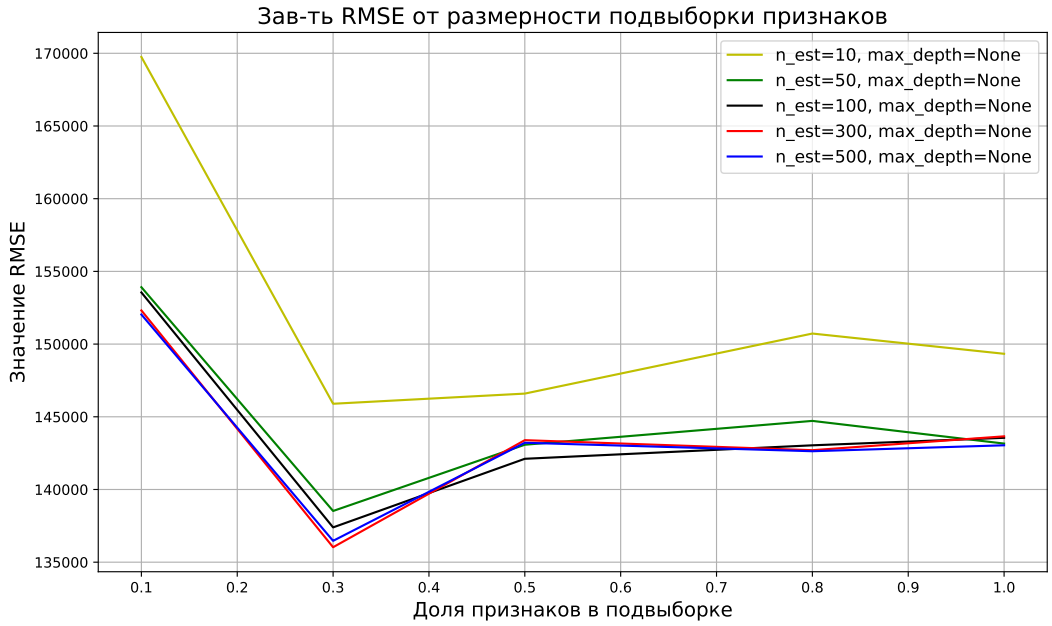


Рис. 6: Зав-ть **RMSE** от размерности подвыборки признаков. Приведены графики при неограниченной максимальной глубине деревьев и разных значениях кол-ва деревьев. Из рисунка видно, что наименьшее значение для всех кол-в деревьев в ансамбле достигается при доле признаков, равной 0.3. Это интересное наблюдение учитывается в псевдокоде **случайного леса** с лекций: в качестве значения по умолчанию для доли признаков в подвыборке используется 1/3.

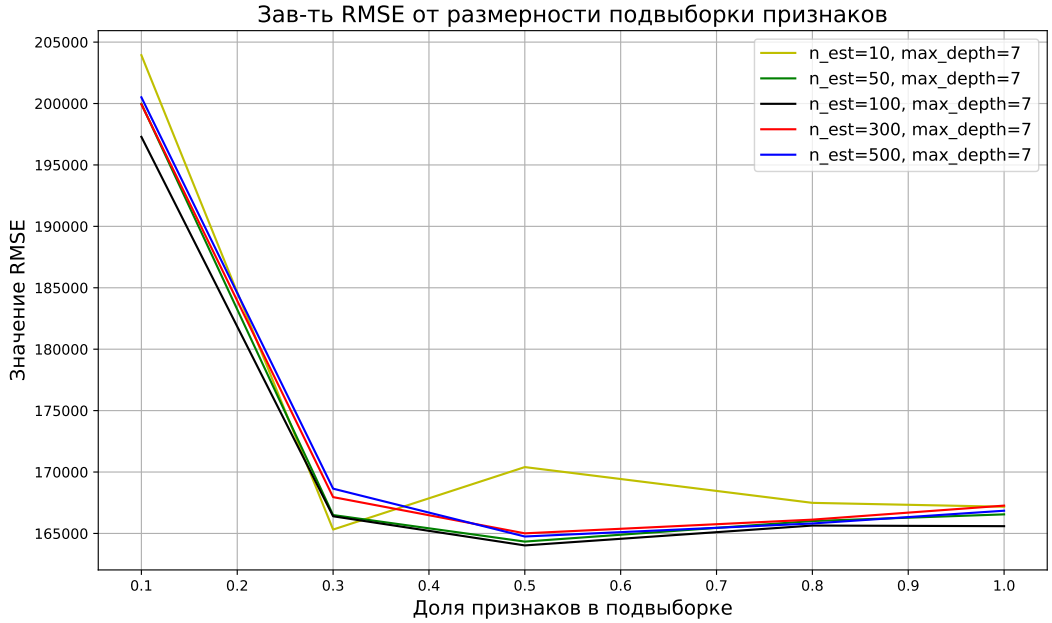


Рис. 7: Зав-ть **RMSE** от размерности подвыборки признаков. Для сравнения со случаем выше приведены графики при максимальной глубине деревьев, равной 7, и разных значениях кол-ва деревьев. В этом случае 0.3 уже не является оптимальным значением доли признаков в подвыборке. Оптимальное значение равно 0.5 при достаточном кол-ве деревьев.

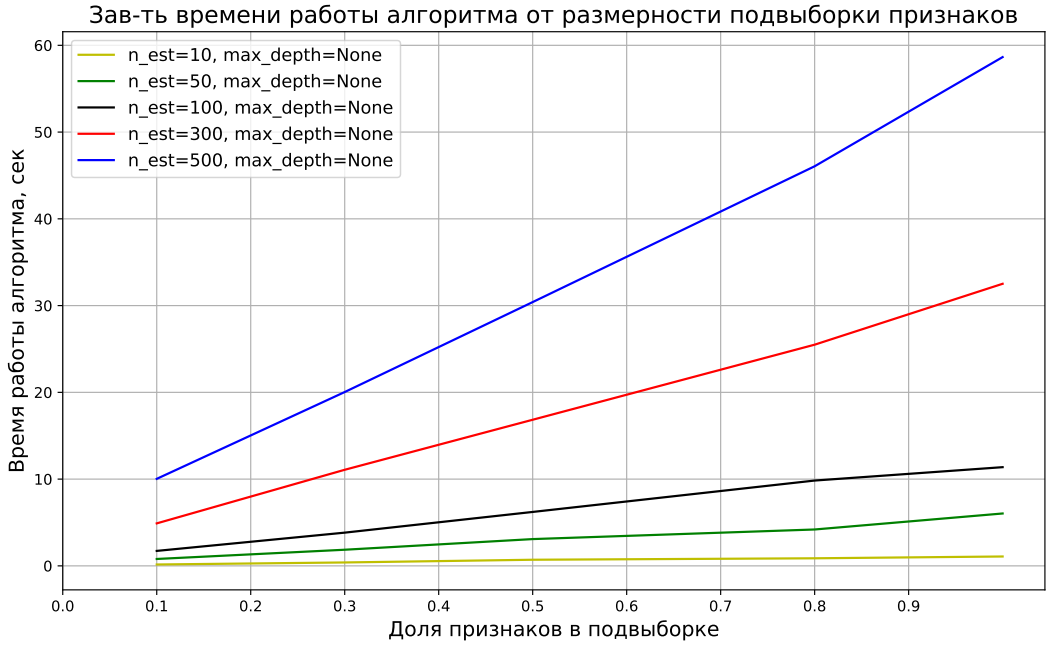


Рис. 8: Зав-ть времени работы алгоритма от размерности подвыборки признаков. Приведены графики при неограниченной максимальной глубине деревьев и разных значениях кол-ва деревьев. Видно, что с увеличением объёма подвыборки увеличивается время работы алгоритма.

Лучший **RMSE**: 136029.

Лучшие параметры:

- количество деревьев - 300;
- размерность подвыборки признаков для одного дерева - 0.3;
- максимальная глубина дерева - неограниченная.

3.3 Выводы из эксперимента

Из проведённых экспериментов видно, что:

- увеличение кол-ва деревьев не приводит к переобучению;
- увеличение максимальной глубины дерева (вплоть до неограниченной глубины) не приводит к переобучению;
- использование всей выборки при составлении очередного дерева не является самым оптимальным вариантом.

Интерпретировать некоторые закономерности позволяет **BVD** разложение:

Для случая простого голосования смещение ансамбля совпадает со смещением базового алгоритма, поэтому увеличение кол-ва базовых алгоритмов никак не влияет на компоненту смещения. При этом увеличение глубины отдельных деревьев уменьшает смещение для всего ансамбля. Разброс же состоит из двух компонент: дисперсии и ковариации. При увеличении кол-ва базовых алгоритмов дисперсия уменьшается со скоростью $1/T$, где T - кол-во базовых алгоритмов. Однако неограниченному увеличению кол-ва базовых алгоритмов мешает ковариация, которая выражает степень коррелированности базовых алгоритмов. Основная задача состоит в одновременном уменьшении дисперсии путём увеличения кол-ва базовых алгоритмов и отборе алгоритмов, которые были бы наиболее некоррелированы между собой. Увеличение максимальной глубины дерева как раз решает проблему отбора алгоритмов: отдельные алгоритмы получаются переобученными - огромный разброс при небольшом смещении, из-за чего алгоритмы, обученные на одних и те же данных, очень не похожи друг на друга, следовательно слабо коррелированы.

4 Исследование поведения алгоритма градиентный бустинг

4.1 Дизайн эксперимента

Исследуем поведение алгоритма **градиентный бустинг**. Изучим зависимость **RMSE** на отложенной выборке и время работы алгоритма в зависимости от следующих факторов:

- количество деревьев в ансамбле;
- размерность подвыборки признаков для одного дерева;
- максимальная глубина дерева (с дополнительным случаем неограниченной глубины);
- выбранный `learning_rate` (каждый новый алгоритм добавляется в композицию с коэффициентом $\alpha \cdot$ `learning_rate`).

Кроме параметров функции $\_\_init\_\_()$, описанных в шаблоне, отдельно обрабатываем параметры `random_state` (значение по умолчанию равно 42) и `continuity` (значение по умолчанию равно False), которые вычленяются из аргумента `trees_parameters`. Всё аналогично **случайном лесу**.

Для экспериментов выбраны следующие значения гиперпараметров:

- количество деревьев в ансамбле: 1, 5, 10, 20, 50, 100, 200, 300;
- размерность подвыборки признаков для одного дерева: 0.1, 0.3, 0.5, 0.8, 1.0;
- максимальная глубина дерева: 1, 3, 5, 7, 10, 15, None (неограниченная глубина);
- `learning_rate`: 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}.

4.2 Результаты эксперимента

Общее время обучения на выбранных выше параметрах составило 57 минут.

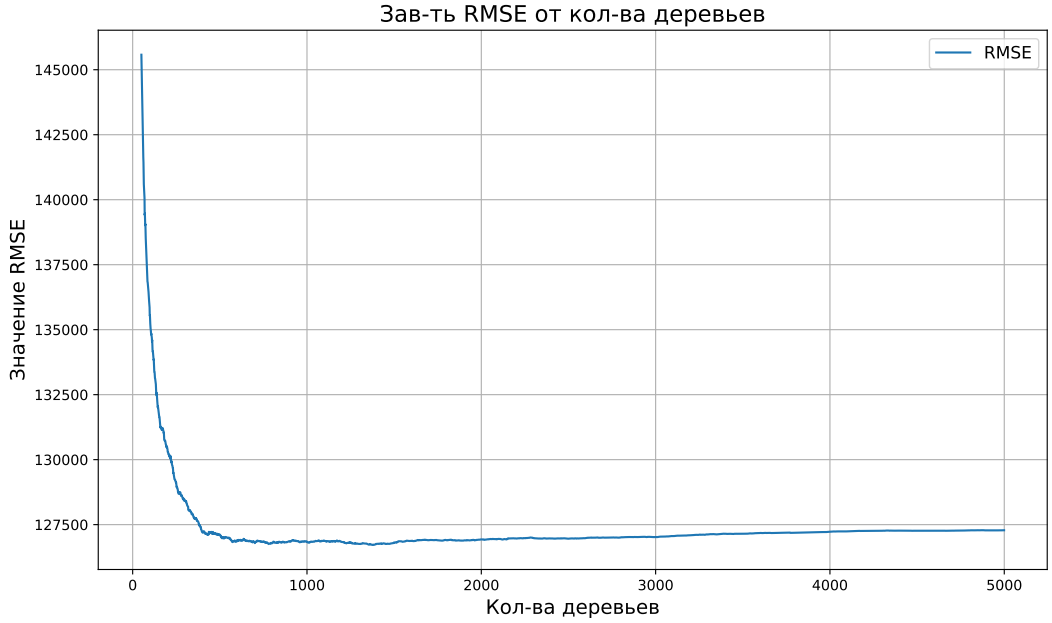


Рис. 9: Зав-ть **RMSE** от кол-ва деревьев. Как видно из графика даже при таком большом кол-ве деревьев в композиции (5000) переобучение очень слабое. При этом есть достаточно большой промежуток, в котором **RMSE** принимает одинаковые значения. В данном случае максимальная глубина дерева равна 5, а объём используемого признакового пространства - 1/3 (значения по умолчанию). Значения **RMSE** выводятся для кол-в деревьев, начиная с 50, для лучшего осознания масштаба на интересующих значениях.

Здесь и далее ограничимся 300ми деревьями в композиции для ускорения постановки экспериментов.

Также на приводимых ниже рисунках зависимостей будут присутствовать графики для разных значений параметров также как и для **случайного леса**.

Убедившись в разумности значений по умолчанию для **случайного леса**, здесь и далее будем использовать значения по умолчанию при демонстрации графиков для **градиентного бустинга** (0.3 вместо 1/3 для доли признаков в подвыборке).

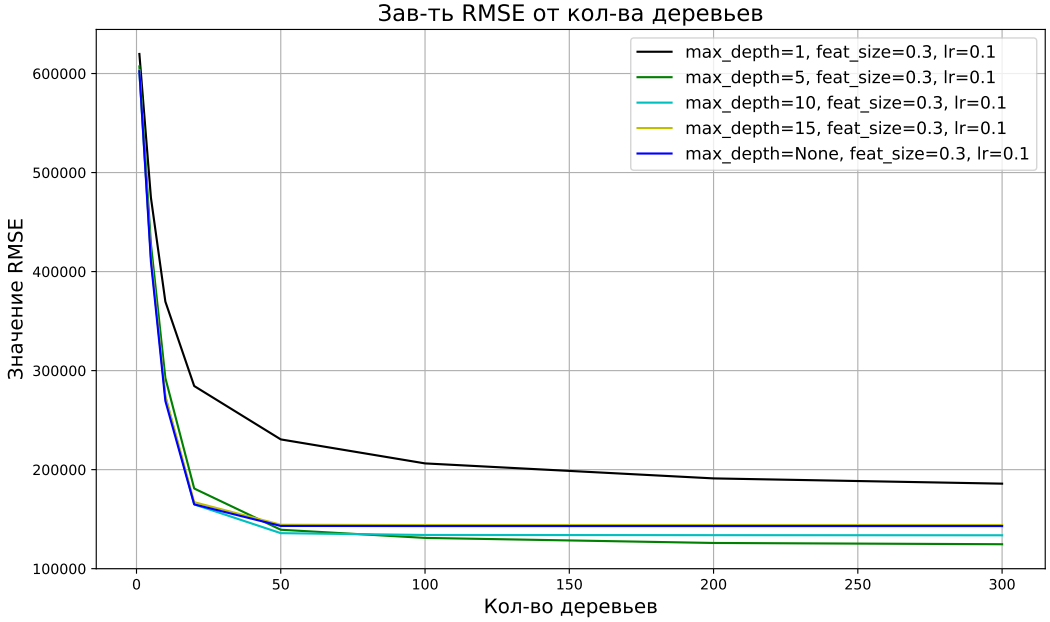


Рис. 10: Зав-ть **RMSE** от кол-ва деревьев. В данном случае приведены графики при доле признаков, равной 0.3, темпе обучения, равном 0.1, и разных значениях максимальной глубины деревьев. Как и на предыдущем рисунке значение **RMSE** уменьшается при увеличении кол-ва деревьев (на рассматриваемом промежутке кол-ва деревьев).

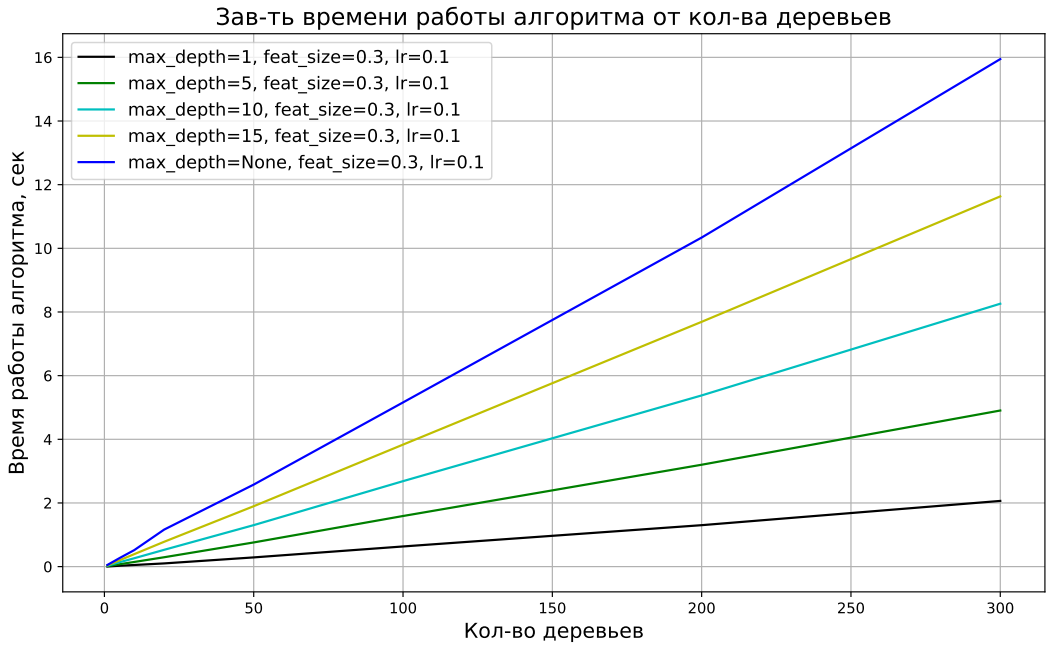


Рис. 11: Зав-ть времени работы алгоритма от кол-ва деревьев. В данном случае приведены графики при доле признаков, равной 0.3, темпе обучения, равном 0.1, и разных значениях максимальной глубины деревьев. Как и ожидалось, увеличение кол-ва деревьев в ансамбле увеличивает время работы алгоритма.

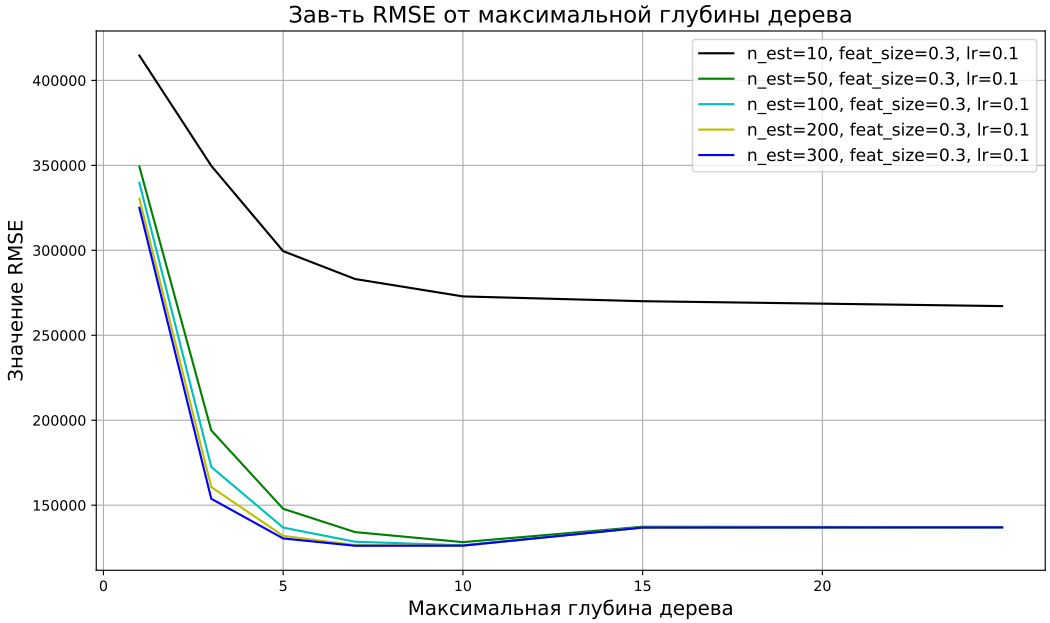


Рис. 12: Зав-ть **RMSE** от максимальной глубины дерева. В данном случае приведены графики при доле признаков, равной 0.3, темпе обучения, равном 0.1, и разных значениях кол-ва деревьев. В отличие от **случайного леса**, неограниченная глубина дерева не является оптимальным вариантом для данного метода ансамблирования. Максимальная глубина, равная 5, является здесь оптимальным значением.

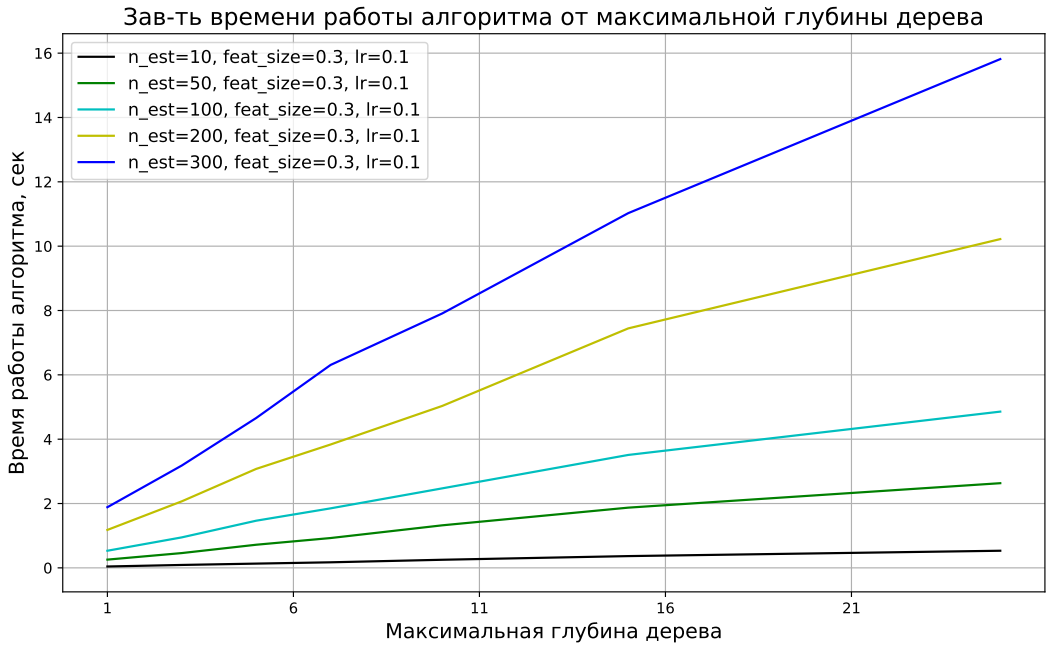


Рис. 13: Зав-ть времени работы алгоритма от максимальной глубины дерева. В данном случае приведены графики при доле признаков, равной 0.3, темпе обучения, равном 0.1, и разных значениях кол-ва деревьев. Как и ожидалось, при увеличении максимальной глубины дерева в ансамбле, увеличивается время работы алгоритма.

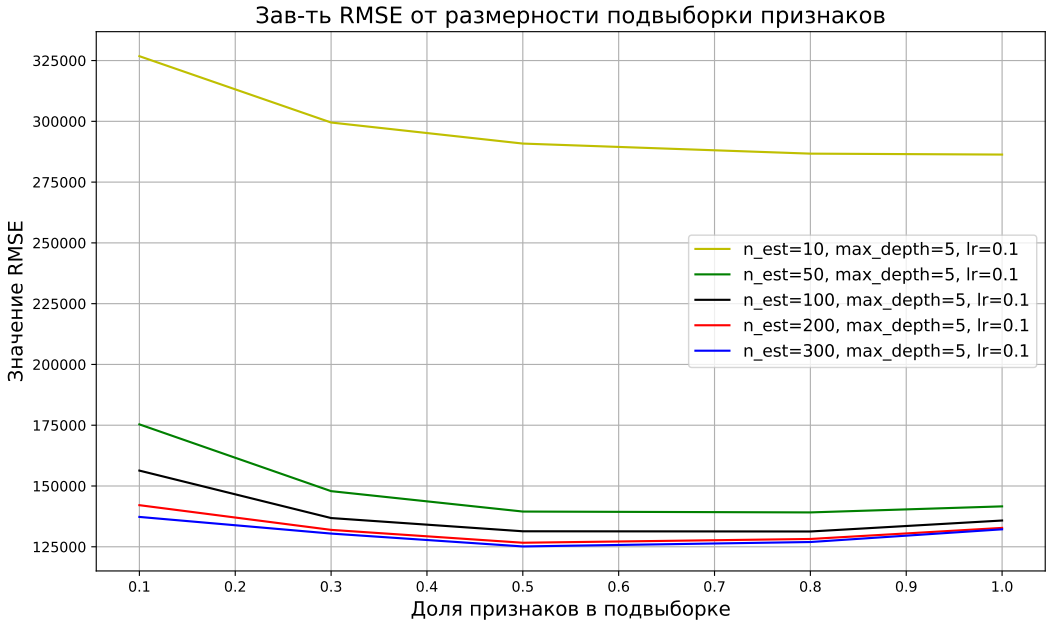


Рис. 14: Зав-ть **RMSE** от размерности подвыборки признаков. В данном случае приведены графики при максимальной глубине, равной 5, темпе обучения, равном 0.1, и разных значениях кол-ва деревьев. Оптимальное значение, равное 0.5, отличается от такового для **случайного леса**.

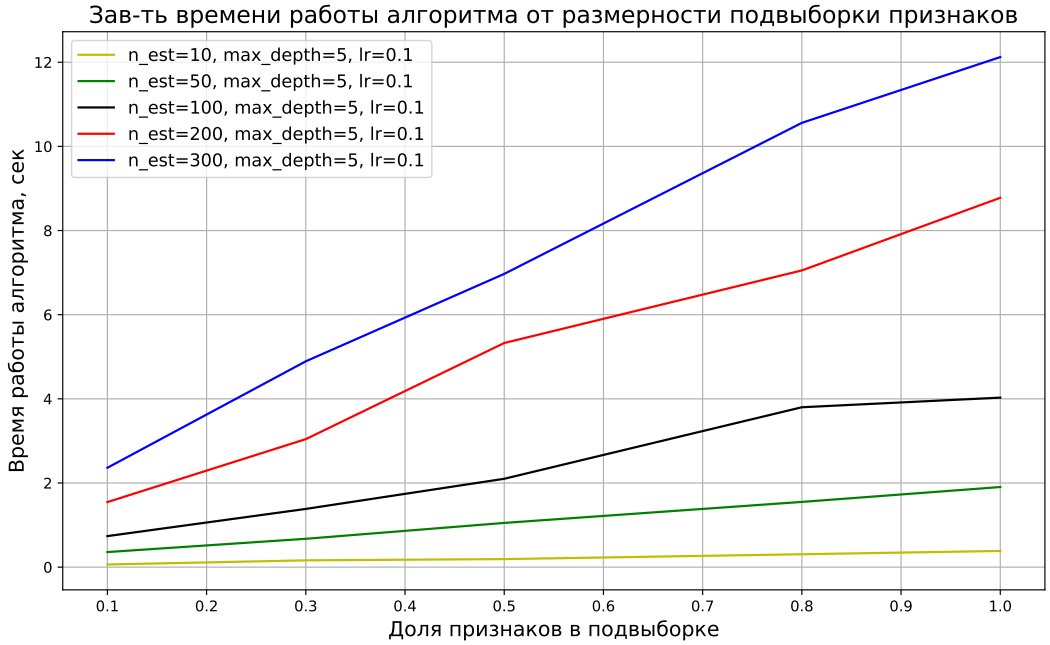


Рис. 15: Зав-ть времени работы алгоритма от размерности подвыборки признаков. В данном случае приведены графики при максимальной глубине, равной 5, темпе обучения, равном 0.1, и разных значениях кол-ва деревьев. Время работы алгоритма увеличивается с увеличением объёма подвыборки.

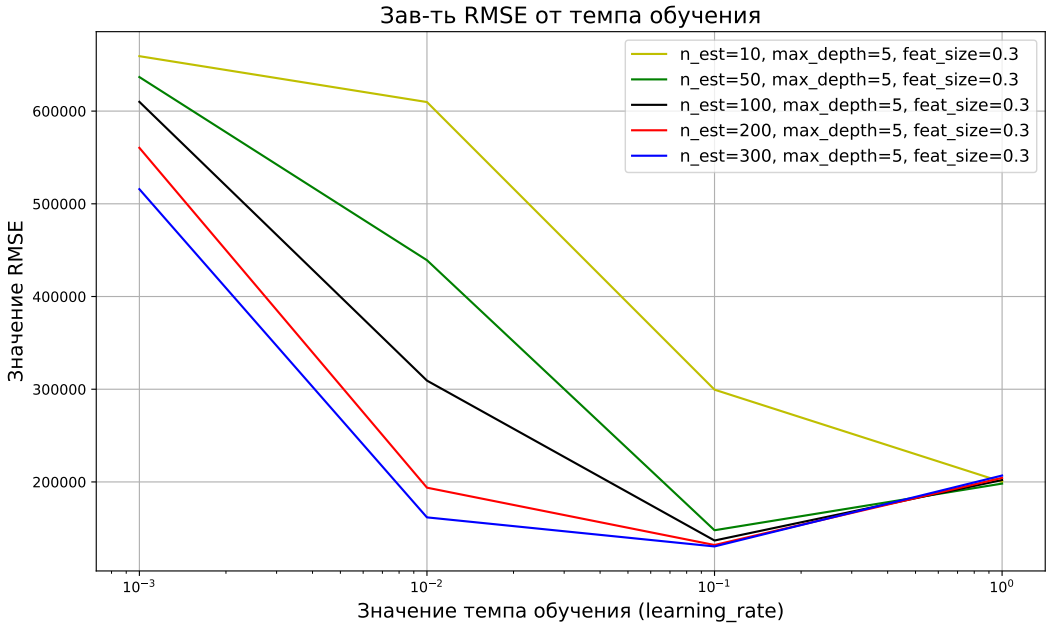


Рис. 16: Зав-ть **RMSE** от темпа обучения. В данном случае приведены графики при доле признаков, равной 0.3, максимальной глубине, равной 5, и разных значениях кол-ва деревьев. Оптимальное значение темпа обучения равно 10^{-1}.

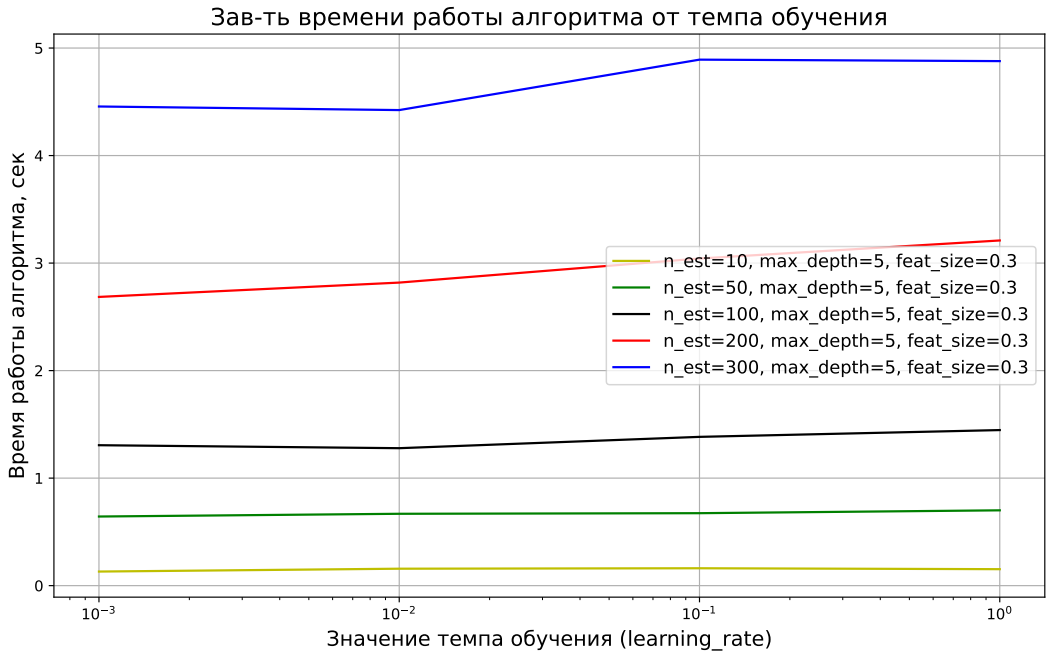


Рис. 17: Зав-ть времени работы алгоритма от темпа обучения. В данном случае приведены графики при доле признаков, равной 0.3, максимальной глубине, равной 5, и разных значениях кол-ва деревьев.

Лучший **RMSE**: 125139.

Лучшие параметры:

- количество деревьев в ансамбле - 300;
- размерность подвыборки признаков для одного дерева - 0.5;
- максимальная глубина дерева - 5;
- темп обучения - 0.1.

4.3 Выводы из эксперимента

Из проведённых экспериментов видно, что:

- неограниченное увеличение кол-ва деревьев приводит к переобучению композиции (хотя это и не так отчётливо заметно на выбранных выше параметрах);
- максимальная глубина дерева должна быть компромиссным решением между небольшим смещением и небольшим разбросом;

- использование всей выборки при составлении очередного дерева не является самым оптимальным вариантом;
- использование отличного от 1 темпа обучения положительно сказывается на итоговом значении качества.

Интерпретировать эти закономерности позволяет всё то же **BVD** разложение:

Градиентный бустинг в отличие от **случайного леса** целенаправленно снижает смещение ансамбля алгоритмов и ничего не делает с их разбросами. То есть разброс итоговой композиции получается не меньше, чем разброс входящих в него алгоритмов. Таким образом алгоритмы, входящие в **градиентный бустинг** должны иметь относительно небольшой разброс. На практике максимальную глубину дерева выбирают равной в промежутке от 3 до 7.

Также нужно отметить, что **градиентный бустинг**, в отличие от **случайного леса**, может переобучаться при большом кол-ве базовых алгоритмов из-за возрастающего разброса, поэтому неограниченно увеличивать кол-во деревьев вообще говоря нельзя.

Использование темпа обучения позволяет точнее уменьшать ошибку композиции, когда разница уже не столь существенна для исходного темпа обучения, однако всё ещё большая.

5 Вывод

Проведены необходимые эксперименты, исследовано поведение различных ансамблей алгоритмов, усвоены основные положения ансамблевых методов обучения, получены необходимые практические навыки.

Список литературы

[1] *Воронцов К. В.* Линейные ансамбли. —

`http://www.machinelearning.ru/wiki/images/3/3a/`

`Voron-ML-Compositions1-slides.pdf` - 2021.

[2] *Воронцов К. В.* Продвинутые методы ансамблирования. —

`http://www.machinelearning.ru/wiki/images/2/21/`

`Voron-ML-Compositions-slides2.pdf` - 2021.

[3] *Соколов Е. А.* Бэггинг, случайные леса и разложение ошибки на смещение и разброс (лекция №9 ФКН ВШЭ) - 2021.

[4] *Соколов Е. А.* Градиентный бустинг (лекция №10 ФКН ВШЭ) - 2021.